TRANSFERRING KNOWLEDGE TO SMALLER NETWORK WITH CLASS-DISTANCE LOSS

Seung Wook Kim & Hyo-Eun Kim
Lunit Inc.
Seoul, South Korea
{swkim,hekim}@lunit.io

ABSTRACT

Training a network with small capacity that can perform as well as a larger capacity network is an important problem that needs to be solved in real life applications which require fast inference time and small memory requirement. Previous approaches that transfer knowledge from a bigger network to a smaller network show little benefit when applied to state-of-the-art convolutional neural network architectures such as Residual Network trained with batch normalization. We propose class-distance loss that helps teacher networks to form densely clustered vector space to make it easy for the student network to learn from it. We show that a small network with half the size of the original network trained with the proposed strategy can perform close to the original network on CIFAR-10 dataset.

1 INTRODUCTION

Neural network models have achieved state-of-the-art performances in multi-disciplines such as in computer vision, automatic speech recognition, and machine translation. It is widely known that models with more capacity (i.e more parameters or more computations) perform better when there is a suitable amount of data. However, it is desirable to have a small model that can work as well as a larger model since it reduces the inference time and physical memory space the model occupies.

The knowledge learned by a well-performing large model can be useful (Hinton et al., 2015). For example, the knowledge that dog is more similar to cat than car can be useful in training a model because the loss function can be constructed so that it penalizes worse predictions more. Often, this knowledge is not available in classification problems where each data x is paired with corresponding label y. Therefore, instead of directly using y, small models can benefit from learning with the knowledge that trained bigger models can provide. This paper proposes a new strategy of transferring knowledge learned by a large network to a smaller network.

2 RELATED WORKS

Approaches to constructing smaller models can be broadly categorized into two classes that are independent: mimic learning and model compression. Therefore, our proposed mimic learning approach can be combined with existing model compression techniques to make models even smaller.

2.1 MIMIC LEARNING

Mimic Learning refers to the strategy of training a student network to mimic the behavior of a pre-trained teacher network. Bucilă et al. (2006) transferred knowledge from an ensemble of models to a single model. Ba & Caruana (2014) and Hinton et al. (2015) trained a student network by matching the logits of the student network and a pre-trained teacher network. Hinton et al. (2015) showed that the small network performed better when it was trained with both the traditional cross-entropy loss and the new objective than when it was trained with only cross-entropy loss. Romero et al. (2015) introduced FitNets that extended this idea to deeper networks by introducing additional hint loss to intermediate hidden layers and showed that this strategy can work in reasonably deep

networks up to 19 layers. However, it did not show much benefit when applied to current state-of-the-art convolutional neural network models such as ResNet with batch normalization (He et al., 2016). Srivastava et al. (2015) trained Highway Network from scratch that had the same structure as FitNets but with additional highway connections, and showed the network performed better than the one that was guided by a teacher network. Chen et al. (2016) also stated they did not find the FitNets approach helpful.

2.2 Model Compression

Model compression techniques aim to directly make sizes of trained model smaller by either reducing the number of parameters or computations. LeCun et al. (1990) used second-derivative information to remove unimportant weights. Jaderberg et al. (2014) sped up the evaluation of CNNs by approximating a learnt full rank filter as combinations of a rank-1 filter basis. Han et al. (2016) reduced the number of parameters using pruning, trained quantization and Huffman coding.

3 Methodology

3.1 Matching Vectors

In classification problems, a neural network, f, typically produces a feature vector $f(x)$, given a data point x, which is mapped to its corresponding label y. This feature vector $f(x)$ entails useful information about x. For example, feature vectors learned by some pre-trained network such as ResNet for the object classification task can be used for other tasks such as visual question answering, image caption generation and object localization.

Recent neural network architectures are highly over-parameterized (Denil et al., 2013) and have deep structures, but these seem necessary to achieve good performance when they are trained with widely used learning signals such as cross entropy loss. These over-parameterized deep structures can be made simpler if there is a better learning signal. Assuming there is a reasonably well-performing teacher network, we plan to use the network to guide a smaller and shallower student network. Previous mimic learning approaches match logits of teacher and student networks to transfer knowledge, but if we can teach a student network to match the feature vectors of its teacher network, the student would be able to behave more like its teacher. Thus, we minimize the $l2$ distance loss L_s between the feature vectors of the student and teacher networks to train $Student$:

$$L_S(x) = ||f(x) - g(x))||_2^2 \tag{1}$$

where f and g represent the teacher and student networks, respectively. The parameters of the teacher are fixed, so only the student's parameters are updated to minimize the loss.

Feature vectors are typically fed into a fully connected layer with weight W to get class probabilities through softmax function. We do not use any additional loss other than L_S. Therefore, $Student$ and $Teacher$ share a fixed W that $Teacher$ has already learned.

3.2 Class-distance loss

Minimizing the $l2$ distance between vectors in a high dimensional space with many data points is not an easy optimization problem. We empirically found that naively minimizing the $l2$ distance converged at an unsatisfactory distance. We propose a new loss that makes the task easier. Let's first define some terms:

$$C_m = \frac{1}{N_m}\sum_{j}^{N_m} f(x^j), \quad C(x) \in \mathbb{S} : \{C_1, C_2, ..., C_k\}, \quad O(x) \in \mathbb{S} \setminus C(x) \tag{2}$$

Suppose there are k classes. C_m is the mean vector of class m with N_m members. $C(x)$ and $O(x)$ map x from class m to C_m and C_n where $n \neq m$, respectively. Now, $Teacher$ is trained so that it minimizes the new loss L_T:

$$L_T(x^i, y^i) = H(\sigma(Wf(x^i)), y^i) + \lambda(||f(x^i) - C(x^i)||_2^2 - min(\phi, ||f(x^i) - O(x^i)||_2^2)) \tag{3}$$

where H refers to the traditional cross entropy loss, σ the softmax function, (x^i, y^i) the i-th data point and its class, λ the weighting parameter, and ϕ the threshold for inter-class distance.

Number of layers	Number of parameters	Baseline	TF-baseline	TF-cdloss
32	0.46M	7.66 (0.21)	7.40 (0.12)	**7.02 (0.10)**
56	0.85M	6.93 (0.08)	6.80 (0.08)	**6.54 (0.15)**
Number of layers	Number of parameters	Baseline	Class-distance loss	
110	1.7M	6.47 (0.13)	6.38 (0.13)	

Table 1: We report the classification error rates of 5 trials with mean (std) on CIFAR-10. $Baseline$ refers to networks trained only with cross entropy loss, $Class\text{-}distance\ loss$ refers to networks trained with L_T, $TF\text{-}baseline$ refers to networks trained by transferring knowledge from 110-layer baseline network, and $TF\text{-}cdloss$ refers to networks trained by transferring knowledge from 110-layer network trained with the proposed strategy.

The loss is designed so that while maintaining the inter-class distance learned by minimizing H, intra-class distance is minimized so that class-wise clusters are more dense. In the high dimensional space, thresholding the inter-class distance prevents the data space from exploding, and having the min operator prevents the data space from contracting to a small region. This will help the student minimize the loss in Section 3.1 since its target vectors are now in more condensed space. The mean vectors are calculated at the end of each epoch as in Expectation-Maximization (EM) algorithm. L_T is related to metric learning (Xing et al., 2002)(Chechik et al., 2009), but the differences to previous approaches are that we use distance to the mean vectors of each class rather than sampling data points in each iteration, and the distances are merely used to shape the feature vector space that is mainly learned by minimizing H. The training process is summarized in Algorithm 1.

Algorithm 1 Training teacher and student networks with class-distance loss

1: **for** $i \leftarrow 1$ to max_epoch **do**
2: train the teacher, f, by minimizing L_T
3: calculate the mean vectors for each class
4: **end for**
5: **for** $i \leftarrow 1$ to max_epoch **do**
6: train the student, g, by minimizing L_S
7: **end for**

4 EXPERIMENTS

We conducted experiments on the CIFAR-10 dataset (Krizhevsky, 2009) that has 10 classes with 50K/10K training/test images. All networks are ResNets with pre-activation units (He et al., 2016). Teacher networks are 110-layer ResNets trained with/without class-distance loss. The teacher networks are used to train smaller student networks with 32 and 56 layers. We used learning rates of $\{0.1, 0.01, 0.001\}$ and $\{0.001, 0.0002, 0.00004\}$ each corresponding to $\{0\text{-}80, 80\text{-}120, 120\text{-}160\}$ epoch for teacher and student networks, respectively. For L_T, ϕ was set to inter-class distance of the baseline network trained only with cross entropy loss, and λ was set to 0 for the first two epoch to allow the network to stabilize and was set to 0.0001 afterwards.

Results are summarized in Table 1. Although the 110-layer network trained with L_T performs similarly to the baseline 110-layer network, when their knowledge is transferred to student networks, the ones that gets knowledge from teacher with class-distance loss clearly outperforms the other models. Appendix includes t-sne plots (van der Maaten & Hinton, 2008) of vectors of each of the network that demonstrate the effectiveness of the proposed strategy.

5 CONCLUSION AND FUTURE WORKS

We showed that training a teacher network with class-distance loss and then training a student network by minimizing the $l2$ distance between feature vectors can improve the performance of the student network. Our preliminary experiments on CIFAR-100 dataset showed reducing the $l2$ distance effective but the effect of class-distance loss was not significant. This suggests that the proposed strategy might be most useful on datasets where the number of classes is small and where the teacher network is highly confident and accurate about its output. Our future work includes tuning the class-distance loss so that it can be readily used on any dataset.

REFERENCES

Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *NIPS*, pp. 2654–2662, 2014.

Cristian Bucilă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. *KDD*, 2006.

Gal Chechik, Uri Shalit, Varun Sharma, and Samy Bengio. An online algorithm for large scale image similarity learning. *NIPS*, 2009.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. *ICLR*, 2016.

Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando de Freitas. Predicting parameters in deep learning. *NIPS*, 2013.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *ICLR*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *ECCV*, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *BMVC*, 2014.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

Yann LeCun, John S. Denker, and Sara A. solla. Optimal brain damage. *NIPS*, 1990.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *ICLR*, 2015.

Rupesh Kumar Srivastava, Klaus Greff, and Ju rgen Schmidhuber. Highway networks. *Deep Learning Workshop, ICML*, 2015.

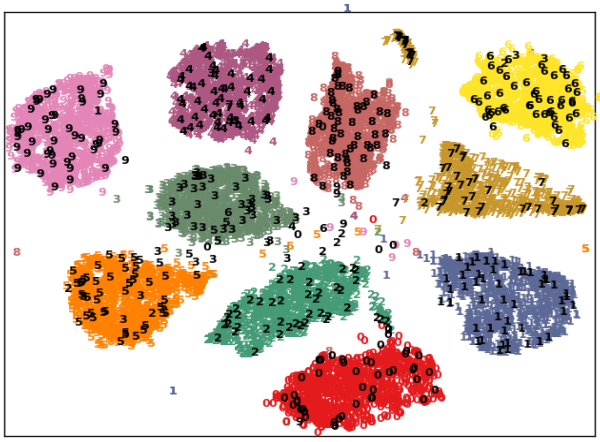
Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008.

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. *NIPS*, 2002.

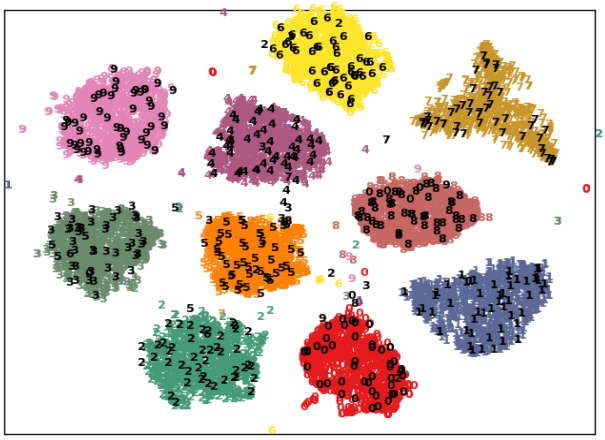
6 APPENDIX

Here we include the 2-D visualization of feature vectors learned by networks using t-sne technique (van der Maaten & Hinton, 2008). Each number on the plot represents a class where black numbers are 50 sampled vectors from the test set and the colored rest are 10,000 sampled vectors from the training set.

6.1 T-SNE PLOTS FOR TEACHER NETWORKS



(a) 110-layer teacher network trained with cross entropy loss only

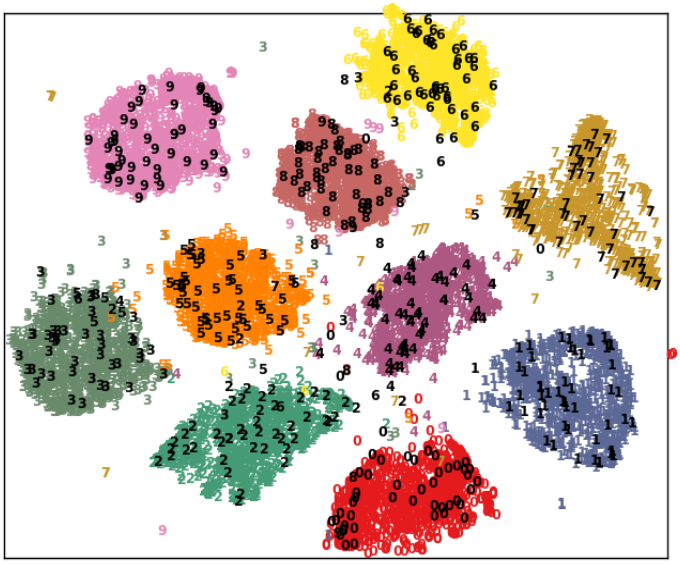


(b) 110-layer teacher network trained with the proposed loss, L_T

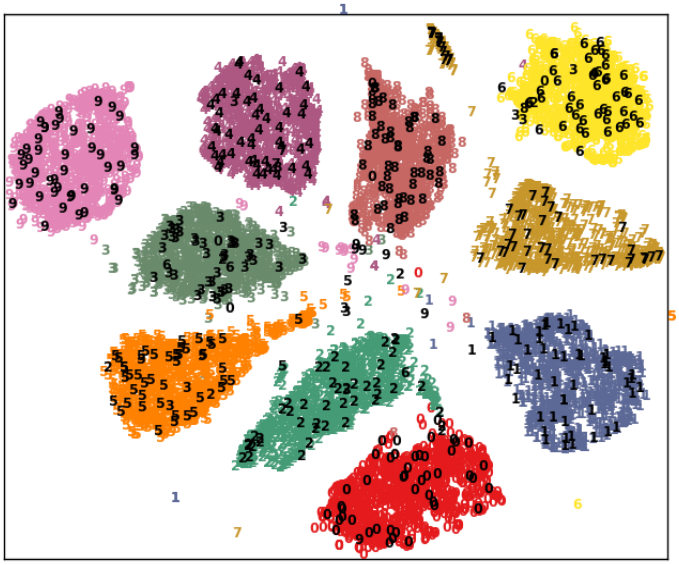
Figure 1: Teacher networks. Even though the two results in similar error rates, (b) has more dense clusters and there are less data points lying on the class boundaries. This makes it easier for student networks to learn feature vectors of (b) than (a).

6.2 T-SNE PLOTS FOR STUDENT NETWORKS

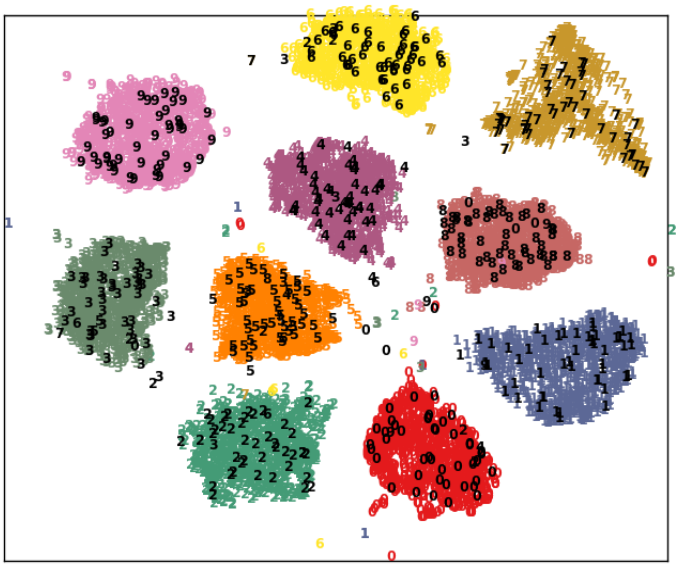
Here, we visualize how learned vectors fit on the teacher networks's vector space. Colored samples are training vectors from the teacher networks while black samples are test vectors learned by the student networks.



(a) 56-layer baseline network trained with cross entropy loss only



(b) 56-layer student network taught by teacher trained with cross entropy loss only



(c) 56-layer student network taught by teacher trained with the proposed loss, L_T

Figure 2: 56-layer baseline and knowledge-transferred networks. By looking at the distributions of test data points in (b) and (c), we can see that the student has successfully learned teacher's vector space. Consequently, (c) has less vectors lying on the class boundaries, resulting in more confident and accurate inference than (a) and (b).